Brian Roche-Campos

CS-300 DSA: Analysis and Design

Southern New Hampshire University

Completed Pseudocode

6/2/2023

// Vector pseudocode

int numPrerequisiteCourses(Vector<Course> courses, Course c) {

totalPrerequisites = prerequisites of course c

for each prerequisite p in totalPrerequisites

add prerequisites of p to totalPrerequisites

print number of totalPrerequisites

}

void printSampleSchedule(Vector<Course> courses) {

for all courses

print course name

if course has prerequisites

for each prerequisite

print prerequisite

}

void printCourseInformation(Vector<Course> courses, String courseNumber) {

for all courses

if the course is the same as courseNumber

print out the course information

for each prerequisite of the course

print the prerequisite course information

}

// Hashtable pseudocode

int numPrerequisiteCourses(Hashtable courses, Course c) {

totalPrerequisites = Hashtable[c]

for each prerequisite p in totalPrerequisites

add prerequisites in Hashtable[p] to totalPrerequisites

print number of totalPrerequisites

}

void printSampleSchedule(Hashtable courses) {

for all key, value pair in courses

print key course name

if value has prerequisites

for each prerequisite

print prerequisite

}

void printCourseInformation(Hashtable courses, String courseNumber) {

for all courses

if the course is the same as courseNumber

print out the course information

for each prerequisite of the Hashtable[course]

print the prerequisite course information

}

// Tree pseudocode

int numPrerequisiteCourses(Tree courses, Node c) {

totalPrerequisites = left and right child of Node c

for each prerequisite p in totalPrerequisites

add left and right Nodes of node p to totalPrerequisites

print number of totalPrerequisites

}

void printSampleSchedule(Tree courses) {

for all Nodes as courses

print course name

if course has left node

print left node as prerequisit

if course has right node

print right node as prerequisit

}

void printCourseInformation(Tree courses, String courseNumber) {

for all Nodes

if the course is the same as courseNumber

print out the node's information

if course has left node

print left node as prerequisite couse information

if course has right node

print right node as prerequisite couse information

end Function

else

if course has left node

goto left node

if course has right node

goto right node

}